

1 Objectivos

Utilização da linguagem Java e dos tipos abstratos de dados leccionados e implementados pelos alunos, para implementar um corrector ortográfico. O aluno deve providenciar juntamente com os ficheiros pedidos a implementação de Tabelas de dispersão, tal como foram tratadas nas aulas práticas. Deve apresentar a implementação de HashTable, LinearHashTable, QuadraticHashTable, DoubleHashTable, que implementou nas aulas práticas. Para a construção da tabela de dispersão que contem o dicionário deve usar uma das referidas tabelas de dispersão (linear, quadrática ou de dispersão dupla), para a tabela que contem as sugestões deve usar uma tabela de tipo diferente ao que usou para o dicionário. Use o *hashCode()* das palavras, como função de *hash* em ambos os casos.

2 O Trabalho

2.1 Descrição

Pretende-se a geração dum corrector ortográfico, que faça uso de tabelas de dispersão. Se a palavra é encontrada no dicionário, assume-se que está correcta, caso contrário assume-se que está mal escrita. O desempenho do corrector pode ser melhorado, guardando numa outra tabela de hash, as palavras incorrectas já encontradas e a sugestão de correcção. Neste trabalho pretende-se que construa um corrector ortográfico, que receba um texto, identifique as palavras incorrectas no texto (i.e. mal escritas) e que faça sugestões de correcção. O output do trabalho deverá ser um ficheiro de texto que contem as palavras mal escritas do documento original e as sugestões de correcção, uma por linha.

2.2 Realização-Implementação da classe SpellChecker

São fornecidos para a realização do trabalho dois ficheiros:

- um, com o texto a corrigir em português
- outro, um dicionário de palavras em português. O dicionário é dado num ficheiro, uma palavra por linha.

Classe *SpellChecker*: Deve providenciar a implementação desta classe. É obrigatório pelo menos um constructor: `SpellChecker(String dictFile, String errorFile, String docFile)` sendo:

- **dictFile** - o ficheiro que contem o dicionário de palavras
- **errorFile** - o ficheiro que contem os erros encontrados e usado para as sugestões. Este ficheiro poderá ter o formato que entender mas um par (palavra mal escrita → correcção) por linha no ficheiro, é perfeitamente aceitável
- **docFile** - o ficheiro a ser corrigido

A sua classe deve possuir um método *void spellCheck(String DocFile)*. Este método deverá ser responsável por correr o ficheiro *docFile*, e gerar uma tabela de dispersão com os erros e sugestões possíveis. O tratamento do ficheiro deve proceder-se do seguinte modo: Para cada palavra do ficheiro, verifique-se se existe no dicionário, se sim não fazer nada, senão veja-se se existe na tabela de sugestões, se sim, não faz nada (já lá estão as sugestões), se não adicione-se à tabela a palavra mal escrita e as sugestões possíveis. As sugestões possíveis são geradas pelas seguintes regras:

1. Adicionar um caracter à palavra
2. Remover um caracter da palavra
3. Trocar caracteres adjacentes.

Se após a aplicação de qualquer das regras a palavra obtida existe no dicionário das palavras, então adicione-se à tabela de sugestões, na entrada correspondente à palavra mal escrita, a nova sugestão de correcção. A sua classe deverá também apresentar o método *void geraFicheiro(HashTable<?> sugestoes, String errorFile)* que gera o ficheiro com as sugestões para os erros detectados no ficheiro, i.e o output do trabalho.

2.3 Entrega

O trabalho deverá ser entregue até ao dia 26 de Janeiro de 2017, sendo realizada a submissão pelo moodle, nos moldes habituais. Deve submeter os seguintes ficheiros:

- A referida implementação de tabelas de dispersão. Estas implementações deverão seguir as sugestões dadas nas aulas práticas(uma classe abstracta *HashTable* e todas as subclasses de endereçamento aberto que implementam os 3 acessos dados).
- A classe *SpellChecker*. Esta classe deverá ter os métodos já referidos e todos aqueles que considere fundamentais.
- Um relatório adequado à apresentação do trabalho

estes ficheiros deverão ser "zipados" e submetidos num único ficheiro com o seu número de aluno. O trabalho é individual.